Claim 10 from U.S. Patent No. 6,085,192	Claim Construction[1]	ANDREW/CODA[2]
10. A system comprising:	This term means a system.	Coda is a system: "In this paper we show that disconnected operation in a file system is indeed feasible, efficient and usable." KS at 4
a communications channel through a firewall comprising one of an HTTP port and an SSL port:	The Texas Court defined "'communications channel' to mean 'a medium for transferring information. A communications channel can be a physical or wireless link.'" Order at 12. The Texas Court defined "'firewall' to mean 'software and/or hardware for protecting an organization's network against external threats, such as hackers, coming from another network, such as the Internet.'" Order at 14-15. Thus, this term means a medium for transferring,	Coda discloses a communications channel through a firewall. Coda discloses a communication channel between a client and one or more servers: "Each Coda client has a local disk and can communicate with the servers over a high bandwidth network" (KS at 4) Fig. 1 also illustrates communications networking joining servers and clients, thereby showing a communications channel. KS at 6. Coda discloses a firewall. Coda uses an authentication server to protect the network. Integrating Security at 251253-254. In "Scalable, Secure, and Highly Available Distributed File Access", IEEE Computer, May 1990, Figure 4 shows a "slave authentication server" controlling access to a file server on the local network. The caption reads in part: "When a user logs in, a client can obtain authentication tokens on the user's behalf from any slave

[1] With respect to the claim terms construed by the Texas Court in the Visto v.Seven Networks case, I have adopted that claim construction.

[2]: The Andrew system, and its successor Coda, was a system that operated in the 1980s and 1990s at Carnegie Mellon University for distributed computing, and especially for the synchronization of files in distributed computing. This system is described in a number of articles. For purposes of this analysis, I will use the following article : J. Kistler & M. Satyanarayanan, "Disconnected operation in the CODA file system", ACM Trans. on Computer Systems, vol. 10 #1, Feb. 1992, pp. 3-25. (referred to as "KS") (a copy is attached as Exhibit D to the Declaration of Stephen Weinstein ("Weinstein Decl."); "Scalable, Secure, and Highly Available Distributed Firel Access," IEEE Computer, May 1990 (a copy is attached as Weinstein Decl., Ex. F; and "Integrating Security in a Large Distributed System, M. Satyanarayanan, ACM Transactions on Computer Systems, Vol. 7, No. 3, August 1989 pp. 247-280 (a copy is attached as Weinstein Decl., Ex. E).

	physically or wirelessly, information through a network protection mechanism implemented via software and/or hardware, where the medium includes a tunnel (encapsulating any desired protocol data unit within an HTTP or HTTPS (SSL) protocol data unit that is granted passage through the network protection mechanism using its designated port). A port is a transport-level address, within a host computer, corresponding to a particular application such as Web browsing using HTTP.	authentication server. The client uses these tokens as needed to establish secure connections to file servers." Coda does not explicitly disclose use of HTTP or SSL to communicate through a firewall. However, HTTP or HTTPS (SSL) tunneling through a firewall was well known to people skilled in the art at the time of this invention. There were public documents explicitly disclosing how to tunnel through a firewall using an SSL implementation. For instance, the Internet draft of Ari Luotonen (Netscape), "Tunneling SSL Through a WWW Proxy", Dec. 14, 1995, available at muffin.doit.org/docs/rfc/tunneling_ssl.html describes one implementation (a copy is attached as Weinstein Decl., Ex. G). On page 2 of this document, which has been publicly available since its publication date, is the following text: "FTP and Gopher get handled by proxies, too. However, this approach has two disadvantages: * The connection between the client and the proxy is normal HTTP, and hence, not secure. This is, however, often acceptable if the clients are in a trusted subnet (behind a firewall). * The proxy will need to have full SSL implementation incorporated into it. This document describes a way to do SSL tunneling without an implementation of SSL, in a way that is compatible with the current WWW proxy protocol (as defined by the HTTP/1.0 specification). The existing implementation of SSL tunneling in the Netscape Navigator and the Netscape Proxy Server conform to this specification. A patch implementing this feature also exists for the CERN proxy server (CERN httpd)." "When SSL tunneling support is put into the SSL source code, it will work for any SSL enhanced application."

		Consequently, communicating through firewall using an HTTP or SSL port would have been an obvious modification to implement in Coda to a person of ordinary skill in the art at the time of the invention. The suggestion to do so would be from Coda itself which discloses one method of communicating through a firewall.
a general synchronization module for operating within the first firewall and for examining first version information to determine whether a first workspace element at a first store has been modified;	The Texas Court defined "'general synchronization module' to mean 'software routines or code that perform the task of determining whether a workspace element and/or an independently modifiable copy thereof (or have) been modified, based on one or more criteria.'" Order at 16. The Texas Court stated that "version information means 'information that can be used to determine the version of a workspace element.'" Order at 26. The Texas Court defined "workspace element" as "'a subset of workspace data such as an e-mail, file bookmark, calendar, or applications program which may include version information.'" Order at 22. The Texas Court defined "the term 'store' as a 'storage location for	As described below, CODA discloses a general synchronization module that operates within the first firewall and examines first version information to determine whether a first workspace element at a first store has been modified. Also, the server monitors and maintains the most current version of a document. As such, the software at the server determines whether a workspace element (an example is a document) has been modified. Coda discloses a synchronization function within the server that compares storeids (to determine whether either a first or second workspace element has been modified) and performs replay operations. "Our use of optimistic replica control means that the disconnected operations of one client may conflict with activity at servers or other disconnected clients. The only class of conflicts we are concerned with are write/write conflicts. ... The check for conflicts relies on the fact that each replica [second workspace element] of an object [first workspace element] is tagged with a storeid that uniquely identifies the last update to it. During phase two of replay, a server compares the storeid [part of the second version information] of every object mentioned in a log entry with the storeid [first version information] of its own replica of the object. If the

	data that may reside on any type of memory device.'" Order at 24.	comparison indicates equality for all objects, the [replay] operation is performed and the mutated objects are tagged with a new storeid specified in the log entry. If a storeid comparison fails, the action taken depends on the operation being validated. In the case of a store of a file, the entire reintegration is aborted. But for directories, a conflict is declared only if a newly created name collides with an existing name, if an object *updated at the client or server* [italics added] has been deleted by the other, or if directory attributes have been modified at the server and the client." (KS at 17). A first examination result is necessarily generated in order to compare the storeids. Modification of the first workspace element is implicit in the possibility of the storeids not indicating equality, e.g., if an object updated at the client or server has been deleted by the other or if directory attributes have been modified. Coda defines replay as a second execution, at the server, of the normal operations that generated changes in workspace elements at the client: "Disconnected operation is a tantalizingly simple idea. All one has to do is to preload one's cache [at the client] with critical data, continue normal operation until disconnection, log all changes made while disconnected and replay them [at the server] upon reconnection." KS at .23. Coda further discloses first and second workspace elements: "It is therefore appropriate to distinguish between first-class replicas on servers, and second-class replicas (i.e., cache copies) on clients." (KS at 7). Coda discloses a first store as data storage

		that a POSA would understand exists in a Unix file server. The Unix file server is cited in the text: "Coda is designed for an environment consisting of a large collection of untrusted Unix clients and a much smaller number of trusted Unix file servers." KS at 4.
a synchronization agent for operating outside the first firewall and for forwarding to the general synchronization module second version information which indicates whether an independently modifiable copy of the first workspace element at a second store on a smart phone has been modified	The Texas Court defined "synchronization agent" to be "software routines or code that sent at least a portion of second version information to a general synchronization module for purposes of synchronization." Order at 25. The Texas Court defined "independently modifiable copy" to mean 'a copy of a workspace element capable of being modified independent of the works pace element. The copy of the workspace element does not have to be in the same format as the workspace element." Order at 17. The Court rejected Visto's proposal that "[t]he copy does not have to be an exact copy." Order at 16-17.	Coda discloses software at the client that forwards version information that indicates whether the independently modifiable copy has been modified. In Coda, there is software named "Venus" that is at the client which provides this information to the server. Coda discloses, as noted in the section above, an independently-modifiable copy (second workspace element) of a first workspace element, second version information, and second examination results. Coda explicitly addresses a file system designed to handle disconnected operation with "powerful, lightweight and compact laptop computers." KS at 7. A "smart phone" is simply an extension of the same "powerful, lightweight and compact laptop computers" described and used in the Coda system. Even if a smart phone is not explicitly disclosed in Coda, extending Coda to include "smart phones" would be obvious to a person of ordinary skill in the art given that the authors of Coda recognize that Coda may be extended to "environments were connectivity is intermittent . . . [including environments] that use wireless technologies such as packet radio." KS at 23. As such, Coda itself provides the suggestion for use of smart phones in its system. Modification of the second workspace element is embodied in the operations (repeated later in replay) carried out by the

		client while disconnected. Coda discloses second version information that consists of the storeid that the second workspace element received at the last synchronization plus the replay information specifying the series of operations on the second workplace element carried out by the client while it was disconnected. "Venus [on the client] records sufficient information to replay update activity when it reintegrates [with the server]. It maintains this information in a per-volume log of mutating operations called a replay log. Each log entry contains a copy of the corresponding system call arguments as well as the version state of all objects [second workspace elements] referenced by the call." (KS at 14). Coda discloses a second store outside the firewall (and thus on a client system): "The performance cost of server replication is kept low by caching on disks at clients ..." (KS at 4).
a synchronization-start module for operating within the first firewall and for initiating the general synchronization module and the synchronization agent when predetermined criteria have been satisfied;	The Texas Court defined "synchronization-start module to mean 'software routines or code which initiate the synchronization process . . . .However, the court does not read the claims or the specification to require, necessarily, that the synchronization-start module be located within a firewall-protected corporate LAN." Order at 26-27.	Coda discloses software at the server for initiating a synchronization where predetermined criteria are satisfied. Coda discloses synch start (replay in the server) triggered by the Venus application in the client sending file id's and the replay log to the server (the predetermined criterion), where the "server performs the replay within a single transaction, which is aborted if any error is detected." (KS at 16) Whether one identifies software within the Venus application in the client that takes the first step, or software within the server that initiates synchronization on the server in response to the request, this meets the requirement of the Texas Court. Because it is at the server, it operates

		within the firewall. Also, it initiates the general synchronization module**.**
means for generating a preferred version from the first workspace element and from the copy by comparing the first version information and the second version information, wherein if only one of the first workspace element and the copy has been modified, then the means for generating selects the one as the preferred version; and	The Texas Court defined "the term 'preferred version' to mean 'a version of a workspace element that is generated or selected from one or more versions.'" Order at 26. The Texas Court stated that the structure for the "means for generating a preferred version from the first workspace element and from the copy by comparing the first version information and the second version information" "is the general synchronization module 425." Order at 30.	Coda discloses an algorithm to compare version information. Coda discloses (see comment in "general synchronization module" section above) that "replay" means recreating at the server the same changes (if any) made (on the second workspace element) at the disconnected client. If such changes of the second workspace element have been made, the preferred version is the new version generated at the client while it was disconnected. If such changes have not been made, the preferred version remains the first workspace element which itself may have changed (as implied in the comment in "general synchronization module" above).
means for storing the preferred version at the first store and at the second store.	The Texas Court stated that the structure for the "Means for storing the preferred version at the first store and the second store" "is the general synchronization module 425." Order at 31.	Coda discloses storing a preferred version at the server and client. Coda discloses cache means for storing its modified second workspace element (which, if it has been modified, becomes the preferred version), and a POSA understands that the server also stores it after replay. "Disconnected operation is a tantalizingly simple idea. All one has to do is to preload one's cache [at the client] with critical data, continue normal operation until disconnection, log all changes made while disconnected [from the server] and replay them [at the server] upon reconnection." (KS at 23).